Noname manuscript No.
(will be inserted by the editor)

Cloud Marginal Resource Allocation: A Decision Support Model

Walayat Hussain · Osama Sohaib · Mohsen Naderpour · Honghao Gao*

Received: date / Accepted: date

Abstract

One of the significant challenges for cloud providers is how to manage resources wisely and how to form a viable service level agreement (SLA) with consumers to avoid any violation or penalties. Some consumers make an agreement for a fixed amount of resources, these being the required resources that are needed to execute its business. Consumers may need additional resources on top of these fixed resources, known as– *marginal resources* that are only consumed and paid for in case of an increase in business demand. In such contracts, both parties agree on a pricing model in which a consumer pays upfront only for the fixed resources and pays for the marginal resources when they are used. A marginal resource allocation is a challenge for service provider particularly small- to medium-sized service providers as it can affect the usage of their resources and consequently their profits. This paper proposes a novel marginal resource allocation decision support model to assist cloud providers to manage the cloud SLAs before its execution, covering all possible scenarios, including whether a consumer is new or not, and whether the consumer requests the same or different marginal resources. The model relies on the capabilities of the user-based collaborative filtering method with an enhanced top-k nearest neighbor algorithm and a fuzzy logic system to make a decision. The proposed framework assists cloud providers manage their resources in an optimal way and avoid violations or penalties. Finally, the performance of the proposed model is shown through a cloud scenario which demonstrates that our proposed approach can assists cloud providers to manage their resources wisely to avoid violations.

Keywords: Cloud computing, cloud resource management, decision support system, marginal resources, fuzzy systems, SLA violation, violation penalty

Walayat Hussain
Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
E-mail: Walayat.Hussain@uts.edu.au

Osama Sohaib
School of Information, Systems and Modelling, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
E-mail: Osama.Sohaib@uts.edu.au

Mohsen Naderpour
School of Information, Systems and Modelling, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia
E-mail: Mohsen Naderpour@uts.edu.au

*Honghao Gao
Computing Center, Shanghai University, Shanghai, China
E-mail: gaohonghao@shu.edu.cn

1. Introduction

Cloud computing offers a broad range of services and a number of opportunities for businesses which choose to adopt this technology. The cloud offers convenient and on-demand access to various computational and storage services. According to a recent press release in April 2019 by Gartner [1], the revenue of the global public cloud services market is projected to grow 17.5% in 2019 to reach a total of US$214.3 billion from US$182.4 billion in 2018. According to this report, more than a third of organizations consider cloud investment as a top three investing priority, because their entire business services and market competition depends on it. We also find that report by Statista [2] published in April 2019 predicts that the size of the global public cloud computing market and its revenue will reach $331.2 billion by 2022. This significant increase in the cloud market raises a variety of challenges for both cloud consumers and providers. One of the key challenges for cloud providers is to ensure the on-time delivery of the services within the agreed QoS parameters as defined in SLA to avoid any violation and penalties.

A service level agreement (SLA) is an important document between a provider and a consumer in any business. An SLA is a legal guarantee between interacting parties which details their liabilities, expectations, obligations, and penalties and joins them in a business relationship for a specified period. One of the primary objectives of service providers is to maximize their revenue/profit by offering and managing their resources in the best possible way. Service providers aim to increase their reputation and trust value by successfully fulfilling their commitment to consumers without SLA violation [3], while a service consumer's goal is to obtain a high quality of service (QoS) as promised by the service provider. Due to the elastic nature of the cloud-computing environment, managing resources and forming a viable SLA is one of the significant challenges for providers, particularly small-scale service providers. SLA negotiation is the first phase of SLA management and if this is dealt with wisely, it can help the service provider avoid violation and penalties. There are a number of methods [4-8], which automate SLA negotiations; however, these approaches are unable to predict the workload for future intervals, which will affect the resource management and the service price to be offered.

Cloud computing provides remote connections to servers and data centers to form a resource management system, supporting a large number of applications and users with different resource demands [9]. Thus, resource management in the cloud networking is one of the key concerns in SLA management. The authors in [10] provide a comprehensive review of resource management in a cloud data centre for applications of economic and pricing models. The authors state that the pricing models were used as a solution to major issues such as bandwidth allocation, request allocation and workflow allocation. The economic and pricing approach generally used for resource management in a cloud environment includes market-based pricing, network utility maximization-based pricing, game theoretic and auction-based pricing. For example, differential pricing [11] is one of the types of market-based pricing that is used to set prices for cloud resources based on the requirements of the consumers. Other pricing models include smart data pricing [12], cost-based pricing [13, 14], Ramsey pricing [15], profit maximization [16], and demand-based pricing [17] to assist cloud providers to maximize their profit and to determine the optimal resource allocation. During SLA negotiation phase, the negotiating parties agree on different Service Level Objectives (SLOs) and QoS parameters for the pricing model for which the consumer pays upfront for a definite amount of resources the required resources and the consumer pays for another set of resources, the marginal resources, only if these are used [18]. However, a consumer receives a guarantee from the service provider that it will reserve not only the required resources but also the marginal resources and will provide these as soon as they are required without any delay. The required resources are those which the consumer uses to execute its business and the marginal resources are those excess resources that a consumer may need due to seasonal demand.

Once the service provider agrees to provide the marginal resources, it is then required to keep these in reserve. These resources are available should the consumer need these, thus maximizing their use of the resources and boosting their profit. When a consumer does not use the reserved marginal resources, it directly affects the service provider in several ways. For example, it prevents other prospective consumers from utilizing these resources. As a result, it affects the service provider's profitability as it is not able to make a sale if the reserved resources are not used. Whereas the cloud provider can earn revenue by utilizing its resources wisely. In line with SLA, the cloud service provider needs to keep the reserved resources irrespective of its usage to avoid violation penalties. The service provider needs an informed decision-making system to form a viable SLA with a consumer by managing its limited resources in the best way and wisely offering marginal resources to a consumer.

In our previous work [19-21], we identified that the consumer's prior history of service violation is strongly involved in the service provider's decision to offer marginal resources to the requesting consumer. A consumer's previous history is used as an input to determine the amount of marginal resources offered to the requesting consumers [20]. The cloud service provider analyzes the consumer record stored at the consumer end required for a new agreement against the QoS parameters and the SLO. However, it is not necessary that the service provider has the consumer's history of the same QoS parameters and SLOs every time the consumer requests a new agreement. With a different set of QoS parameters and SLOs, the service provider finds it difficult to decide whether to offer the marginal resources to a consumer. Therefore, it is imperative to set up intelligent system to assist service providers to make appropriate decisions while forming SLA with consumers.

In this paper, we focus on the second condition of a request in which a consumer has previously interacted with the service provider. A consumer request for the same SLO with the same set of QoS parameters as requested in an earlier request or it may request a different SLO. We address the issue by proposing a decision support model using a user-based collaborative filtering method. We used an enhanced top-k nearest neighbor algorithm and a fuzzy logic system to assist the cloud provider to form a viable SLA. Our proposed system assist the service provider in decision making while offering the marginal resources to a consumer.

The rest of the paper is organized as follows: Section 2 reviews the related literature on cloud SLA management. Section 3 presents the fuzzy decision support model. Section 4 details the implementation and evaluation. Section 5 concludes the paper and provides the future research directions.

2. Background and Related Work

The following section presents some of the related literature on SLA management architecture. We divide the related existing literature into three categories, 1) existing SLA management approaches, 2) the 2-Phase Viable SLA management approach, and 3) the critical analysis of existing approaches with reference to the 2-Phase viable SLA management approach.

2.1. SLA Management Approaches

There are many SLA management approaches [22-27] which propose different frameworks, such as the collaborative filtering method, the CBR approach, self-monitoring, self-healing, risk assessment, neural networks and other mathematical models to detect SLA violation before the actual violation occurs. SLA management comprises many activities, including monitoring, prediction, and risk analysis and once a possible violation has been detected, recommendations are given for appropriate action to avoid service violation. SLA monitoring is one of the key components in SLA management. Most of the literature [25-27] assists the consumer in the proper management of SLA while the other literature [24, 28] supports the service provider. The work in [26, 27, 29, 30] tackled SLA management as an optimization problem using different similarity matching methods such as ontology and semantic technologies - OWL, RDF and SPARQL, in which the focus is to assist the cloud provider to achieve consumer trust and high satisfaction by providing committed QoS in a timely fashion. Our emphasis in this paper is to assist the service provider in forming a viable SLA by managing resources wisely to avoid any discrepancies that cause service violation. Based on different SLA management techniques, we divide the existing approaches into four categories as presented in the following sub-sections:

2.1.1. LoM2HiS prediction approaches

In these types of prediction approaches, the systems convert low-level metrics to different SLA parameters and using these metrics, tries to predict service violations. The authors in [22] proposed a low-level metrics to a high-level *SLA* (LoM2HiS) framework by mapping low-level resource metrics such as downtime, uptime to high-level SLA parameters such as throughput, response time for the better management of an SLA and to avoid violation. The authors used both simple and complex methods for mapping resource metrics to SLA parameters and stored it in the repository which is further accessed by the run-time monitoring module to check the service status. The run-time module compares the resource metrics against the predefined threshold value, and when it detects a service incongruity, it alerts the service provider to take appropriate action. Although the approach detects a service violation, it does not provide a proper mechanism by which to deal with it, moreover, the authors defined only a few rules for the conversion of resource metrics to an SLA parameter and in the case of service violation, it is very difficult to identify the parameter to tackle it. While using the LoM2HiS framework, the authors in [23] proposed the *Detecting SLA Violation infrastructure* (DeSVi) to predict SLA violation and manage SLA. The proposed model monitors on a run-time basis and communicates with a consumer and a provider in the case of any violation detection. The monitoring agent receives a request from a consumer, calculates the related resource metrics, and forwards it to the run-time monitoring module to map it with the metrics defined by LoM2HiS. Although the proposed system manages SLA by predicting service violation, it lacks the ability to provide a proper recommendation in the case of a service violation. In [31], the authors combine three frameworks: LoM2HiS, the hierarchical layer model LAYSI and the rule-based SLA aggregation and validation model. The proposed framework predicts SLA violation and assists a consumer by imposing a penalty on the service provider when it finds the provider is unable to deliver a committed QoS. Although the proposed system assists in terms of imposing a penalty, there is no description of how the problem is solved.

2.1.2. Neighbourhood-based collaborative approaches

The authors in [24] proposed a neighborhood-based collaborative approach to predict QoS and manage SLA. In this work, the authors used both user-based and item-based collaborative filtering methods to predict the QoS for new requesting consumers. The proposed approach operates in different phases. The first phase gathers QoS data from similar users using the Pearson's correlation coefficient and by selecting the top-K nearest neighbors. In the second phase, the QoS is predicted using both the item-based and user-based collaborative filtering method. Although the approach works well for all consumers who request the same set of QoS parameters as requested earlier, this is not the experience in all cases. Moreover, the approach lacks the prediction interval and criteria for monitoring which plays an active role in any prediction. To reduce the chance of error in the collaborative filtering method due to high data sparsity, or ignoring other related information, the authors in [32, 33] proposed a probabilistic matrix factorization (PMF) method to predict a QoS that considers network location and the association between users and services. In the proposed work, the authors clustered all the consumers using the clustering method and based on the clustering results, the PMF model incorporates the implicit association between users and services. The proposed system grouped consumers into a different region based on network location and QoS parameters and used these inputs for the prediction using matrix factorization. The authors in [34] propose a novel collaborative recommendation framework based on matrix factorization and network location-aware neighbor selection. In addition, the authors in [35] suggests a new matrix factorization model with deep features learning by integrating a convolutional neural network to predict higher QoS results and to improve the accuracy of neighbors selection.

2.1.3. SLA management by QoS prediction

The authors in [36, 37] proposed the QoS monitoring as a service (QoS-MONaaS) model to predict QoS parameters. The proposed model monitors the QoS parameters using a stream processing framework that operates on the SRT-15 [38] platform. The framework comprises two tiers: the business logic tier and the data tier which monitors and checks the related QoS parameters and then executes the monitoring algorithm to monitor all QoS parameters and triggers an alarm to the service provider in the case of service violation. Although the approach assists QoS prediction, the authors do not mention the working of the prediction algorithm nor do they mention the necessary action which needs to be taken by the service when a violation has been detected. Moreover, the approach only works when both interacting parties use the SRT-15 platform. The authors in [25] proposed a consumer-oriented QoS monitoring approach that combines two techniques, the extreme value theorem (EVT) and social network analysis (SNA) to predict consumer performance which has similar behaviours. The proposed approach identifies the strength between a consumer and cluster-related consumers in communities that have the same behaviour. The proposed approach assists cloud (IaaS) service providers with better prediction accuracy about a consumer and enhances their service offerings. The authors in [28] evaluated the prediction accuracy of nineteen prediction methods to assist the cloud provider in managing SLA and to avoid violation in the earliest possible time. The authors divided these approaches based on the time series and machine learning algorithms to evaluate the prediction accuracy of each approach on three QoS parameters, namely throughput, response time and availability based on the Amazon EC2 cloud dataset. Although these comparative analyses assist the cloud provider to decide on an optimal prediction method based on varying datasets to manage SLA, it does not describe which of the early possible remedies to take when the prediction approaches predict a service violation. The authors in [39] used a regression-based model to predict the QoS parameters at different checkpoints to predict the numerical value for each SLO. However, the checkpoints do not have any realistic basis and there is no valid proof of how an early prediction can assist the cloud provider or consumer to avoid violation. Moreover, there is no description of the dataset. The authors in [40] proposed a workload analyser method to predict SLA violation by monitoring different resources. The proposed framework forms an outline for complete resource usage of all applications from platform and infrastructure layers and uses different mathematical methods to predict future workload and identify future service violation.

2.1.4. Ontology-based SLA management approaches

Ontology, the semantic web method and the similarity matching approach are widely used methods to manage a cloud SLA that expresses the notion within a region and designates the way that these ideas are correlated to each other into reasonable assumptions formulated in a conventional language. The authors in [29] proposed an ontology-based framework to simplify the procedure of service deployment in the multi-cloud environment and focus on QoS modelling and implementation optimization. The authors used optimization techniques using ontology-based discovery and a deployment descriptor to reduce human intervention in semantic cloud service product matching. The authors proposed additional QoS parameters including the reliability of the service provider, cost and the latency for selecting cloud providers. The authors in [26] proposed SLA matching and a provider's selection model that automatically discovers semantically identical SLA parameters and creates SLA matchings between different syntax parameters. The process of SLA matching comprises three steps that automatically select the service provider using a public SLA template that obtains the top matching value provided by the cloud service provider in cloud computing marketplaces. However, the proposed approach only works for consumers when the equivalence probability value is more than the pre-set baseline. Otherwise, the system reports an inappropriate service in the existing market based on the consumer's requirements. The authors in [27] described SLA ontology and used different semantic web technologies, OWL, RDF and SPARQL, to automate the process of SLA monitoring and management in a cloud environment. The proposed approach extracts SLA metrics from cloud-related legal documents from a public domain such as cloud provider websites, and then the related terms are extracted from the document automatically and mapped into the previously proposed SLA ontology. Although it helps the consumer to monitor SLA performance, this prototype can only store numeric metrics and a limited number of providers are included in the terms of extraction without providing detail on how the approach works in the case of a large set of service providers.

2.2. Viable SLA Management Framework (2Phased-VSLAM)

SLA management comprises different activities such as SLA formation, SLA monitoring, SLA violation prediction, violation risk assessment and risk-based decision-making to avoid SLA violation and penalties. Various approaches in the literature focus on each of these activities in SLA management process, however, scant literature discusses all the activities in one framework and only limited literature discusses SLA management in the pre-interaction time phase [41, 42]. We categorized these activities of SLA management into two phases, pre-interaction and post-interaction. The pre-interaction phase deals with the activities before forming an SLA, and the post-interaction phase activities manage SLA after entering into a contract [19]. The activities of the pre-interaction and the post-interaction phases discussed in detail [43, 44] are presented in Figure 1.

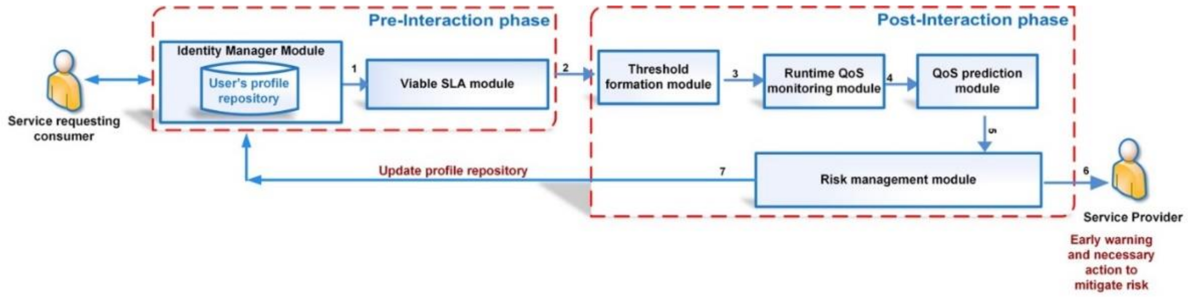

Pre-Interaction phase
Post-Interaction phase
Identity Manager Module
User's profile repository
1
Viable SLA module
2
Threshold formation module
3
Runtime QoS monitoring module
4
QoS prediction module
5
Risk management module
6
7
Update profile repository
Service requesting consumer
Service Provider
Early warning and necessary action to mitigate risk

Figure 1: The 2-Phased SLA management modules.

The process of SLA management starts when a consumer places a request for services to the service provider. The service provider receives a request and validates the consumer using an identity manager module (IMM). The request is forwarded to the viable SLA module (VSLAM) where a consumer's request and its previous transaction history are assessed. Using the calculation result, a decision is made to approve the request or not and if the request is approved, then the number of marginal resources is to be offered to a consumer [18]. Once the service provider determines the resources to offer and a requested consumer agreed on it, then both parties formally generate a SLA.

From this point onward, the process of the post-interaction phase includes all the activities to monitor and predict QoS and assess the risks to take appropriate action. The threshold formation module (TFM) is the first module in the post-interaction phase that defines a threshold value in respect of each SLOs defined in SLA [28]. Mustafa et al. [45] presented two SLA-aware energy-efficient resource management techniques, the available capacity and power (ACP) and the required capacity and power (RCP) which reduce energy consumption while ensuring the agreed QoS level by using the threshold mechanisms. The authors in [22] used case-based reasoning (CBR) approaches by mapping low-level resource metrics to high-level SLA parameters and defining threshold values to compare the deviation between agreed and actual service usage to detect possible service violation. Moreover, the CBR approach has its limitations, such as adaption, processing time, storage and for most, not giving optimal results [46]. When the runtime availability of resources drops below the threshold value, the service provider is alerted to manage the risk of SLA violation in the risk management module (RMM). Risk management is often undertaken through an SLA in a service-oriented context [47]. SLAs also include financial penalties with adverse consequences, such as a customer having signed an agreement with a cloud provider is entitled to compensation in the case of non-compliance, which include QoS, privacy, and the execution environment, etc. To properly manage the risks of SLA violation, the system should have runtime QoS monitoring and prediction modules. After defining the threshold values, the runtime QoS monitoring module (RQoSMM) monitors the runtime QoS parameters and forwards it to the QoS prediction module (QoSPM) where it is used to recalibrate the QoS in the near future [48]. When the system finds that the value of QoSPM exceeds the threshold value, then the risk management module (RMM) is activated to determine the risk of possible SLA violation and take the appropriate action to mitigate it. The detail on the risk management module is presented in [49].

2.3. 2-Phased-VSLAM Existing Approaches

In this section, we analyse the existing approaches with reference to the 2-Phased-VSLAM. We have seen that most of the literature focuses only on one phase of SLA management and very few discuss the decision support system to assist the cloud provider to make an informed decision about the allocation of cloud marginal resources. We compared the existing approaches based on SLA management before SLA execution and after SLA execution, the capability to predict violation, the steps for managing violation and the resource allocation decision for marginal resources. The comparative analysis is presented in Table 1.

Table 1: Critical evaluation of existing SLA management approaches (reproduced from [41]).

Source	SLA administration		Forecast violation	Procedure for handling the violation	Marginal resource management
	Pre-interaction	Post-interaction			
Emeakaroha et al. [22]	✗	✓	✓	✗	✗
Emeakaroha et al. [23]	✗	✓	✓	✗	✗
Brandic et al. [50]	✗	✓	✓	✓	✗
Haq et al. [31]	✗	✓	✓	✓	✗
Romano et al. [36]	✗	✓	✓	✗	✗
Leitner et al.[39]	✗	✓	✓	✗	✗
Hussain et al. [28]	✓	✓	✓	✗	✗
Mustafa et al.[45]	✗	✓	✓	✗	✗
Aazam et al. [10]	✓	✗	✗	✗	✗
Shen et al. [11]	✓	✗	✗	✗	✗
Badidi [14]	✓	✗	✗	✗	✗
Wood et al. [16]	✗	✓	✗	✓	✗
Ciciani et al. [40]	✗	✓	✓	✗	✗
Emeakaroha et al. [51]	✗	✓	✓	✓	✗
Mosallanejad et al. [52]	✗	✓	✗	✓	✗
Katsaros et al. [53]	✗	✓	✗	✗	✗
Sun et al. [54]	✗	✓	✓	✗	✗
Cardellini et al. [55]	✗	✓	✓	✗	✗
Pacheco-Sanchez et al. [15]	✓	✗	✓	✗	✗
Schmieders et al. [56]	✗	✓	✓	✓	✗
Noor and Sheng [57]	✗	✓	✗	✗	✗
Fan and Perros[58]	✓	✗	✗	✗	✗

From the above discussion, it is evident that even though many approaches have been proposed in the literature for cloud SLA management, these do not cover the problem of marginal resource allocation decisions. Some of the shortcomings in the discussed approaches are as follows:

- Most of the literature discusses and proposes an SLA management framework in the post-interaction phase when both parties have finalized and signed off on the execution of their SLA.
- None of the literature considers the consumer's previous history to determine the number of marginal resources offered to a consumer for better service management.
- Scant literature is available to assist the service provider to make a decision about service formation that will result in failing to achieve economic benefit in a given time period.
- Most of the literature discusses SLA management from a consumer's perspective. There are only a few studies that discuss SLA management from a cloud provider's perspective and none of the literature covers all aspects of SLA management, mainly generating negotiation, forming an SLA

(resource allocation for required and marginal resources), monitoring SLA, prediction of violation, estimating risk of violation and alarming the service provider to take early possible remedial action.

In the next section, we propose the fuzzy decision support model to assist the cloud provider in making an informed decision regarding the approval of marginal resource requests.

3. Cloud Marginal Resource Allocation Decision Support Model

The majority of the proposed models focus on the post-interaction time phase and few literatures has discussed marginal resource management, which is an important factor for small-to-medium-scale service providers to avoid SLA violation and penalties [6]. In this study, the 2-Phase-VSLAM is developed further to assist cloud providers in their decision making [19]. To achieve this, a cloud marginal resource allocation decision support (CMRADS) model is developed, as presented in Figure 2.

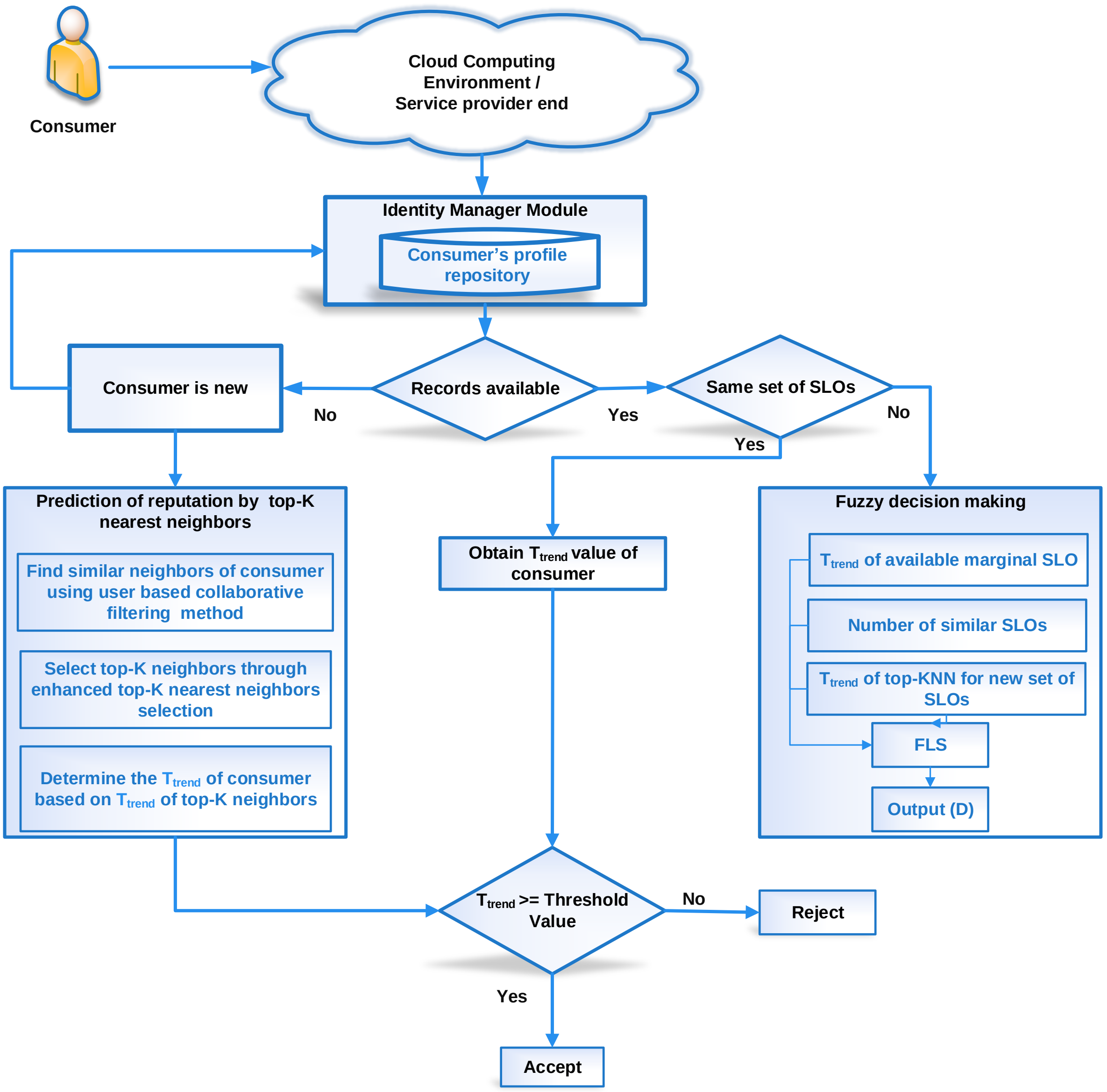

Consumer
Cloud Computing Environment / Service provider end
Identity Manager Module
Consumer's profile repository
Consumer is new
No
Records available
Yes
Same set of SLOs
No
Yes
Prediction of reputation by top-K nearest neighbors
Find similar neighbors of consumer using user based collaborative filtering method
Select top-K neighbors through enhanced top-K nearest neighbors selection
Determine the T_{trend} of consumer based on T_{trend} of top-K neighbors
Obtain T_{trend} value of consumer
Fuzzy decision making
T_{trend} of available marginal SLO
Number of similar SLOs
T_{trend} of top-KNN for new set of SLOs
FLS
Output (D)
T_{trend} >= Threshold Value
No
Reject
Yes
Accept

Figure 2: The cloud marginal resource allocation decision support model.

As shown in Figure 2, when the cloud provider receives a request from a consumer, it searches the consumer in its repository to obtain their history. There are two possible situations: a) the consumer is new and does not have any record with the service provider, or b) the consumer has previously interacted with the service provider. For the former, it is handled by prediction of reputation by top-K nearest neighbours, while for the latter, it is forwarded to the process of using Fuzzy decision marking.

In details, if the consumer is new and there is no record of them in the repository, the service provider can search for the nearest neighbour using the enhanced top-K nearest neighbour selection method [59], the user-based collaborative filtering method and the Pearson correlation coefficient discussed in detail in our previous work [18]. If the consumer has a previous record with the service provider, two possible situations should be considered. The new set of SLOs is the same as previously requested. For example, a consumer requests “storage” and “memory” for marginal resources which is the same as a previous request. The new set of SLOs is different from the previously request. For example, a consumer requests “CPU” and “memory” for marginal resources, whereas it has previously requested “storage” and “memory”. Therefore, the new request is different in one or more parameters.

3.1. Decision Making with the Same SLOs

When the service provider receives a request for marginal resources, it queries the IMM module to retrieve the consumer’s transaction history and the set of SLOs: availability, throughput, response time etc. requested earlier, from a profile repository, as shown in Figure 1. When a consumer has a record of requests for marginal resources with the same set of SLOs as those requested for a new SLA, then the VSLAM calculates the transaction trend - T_{trend} [60] of the requesting consumer to determine whether or not to accept the consumer’s request. The T_{trend} is calculated by taking the ratio of the successful transactions, T_{succ} over the total transactions, T_{total}, whereas T_{total} is the total number of a consumer’s transactions whether or not the marginal resources were used. T_{succ} is all those transactions in which a consumer requested a marginal resource with the same set of SLOs as a new request and used it as well. Mathematically, the T_{trend} is presented in Equation 1.

$$T_{trend} = (\frac{T_{Succ}}{T_{total}}) * 100 \tag{1}$$

The service provider compares the value of T_{trend} with the predefined threshold value. The value of a threshold is a customized value that depends on various factors, the reliability of a consumer, risk attitude of the service provider, and contract duration value [49, 61]. If the value of T_{trend} is equal to or greater than the threshold value, then the service provider accepts the request; otherwise, the request is rejected, as presented in Equation 2.

$$\text{Accept request} = T_{trend} \geq Threshold\ value \tag{2}$$

The reliability of a consumer depends on the commitment of the assured resource usage (marginal resource) by a consumer which increases for every successful commitment or vice versa. A consumer who has previously requested and used marginal resources has a high reliability value compared to a consumer who makes a commitment but did not use it. Based on the reliability of the service provider, we divide them into three categories, Bronze, Silver and Gold [18]. The second factor is the risk attitude of the service provider which indicates its approach to a risk of service violation. We divide the risk attitude of the service provider into three categories, risk valour, risk neutral and risk reluctant. The service provider with a risk reluctant attitude is very cautious about offering any resources to a consumer whose reliability value is low compared to a provider with a risk valour attitude. The last factor that we considered is contract duration which is the time period for which the service provider keeps reserved resources for a consumer. For marginal resources, the service provider always prefers to reserve these for the minimum time.

3.2. Decision Making with Different SLOs

This situation represents an uncertain decision-making problem with a number of uncertain parameters. Therefore, one has to define how the different parameters are combined to draw a conclusion. In this sense, the use of fuzzy logic systems (FLSs) is appropriate. A FLS defines heuristic or expert rules (of the type “IF conditions-THEN conclusions”) to express the relationships between fuzzy parameters (the IF part of the rules) and the outputs or conclusions one can infer from these combinations (the THEN part) [62]. The following definitions provide the preliminary definitions of fuzzy sets that will be used in this section:

Definition 1 (Fuzzy set) [36]: Fuzzy set A is defined in terms of a universal set X by a membership function that assigns to each element $x \in X$ a value $\mu_A(x)$ in the interval [0,1], i.e. $A: X \to [0,1]$.

Definition 2 (Fuzzy number) [36]: A fuzzy number A on $\mathbb{R}$ is a fuzzy subset of the real line where $\mu_A(x)$ denotes the value of the membership function of A in x, satisfying the following properties:

- A is normal, i.e., there is $x_0 \in \mathbb{R}$ such that $\mu_A(x_0) = 1$.
- A is fuzzy convex, i.e., $\mu_A(\lambda x + (1-\lambda)y) \geq min\{\mu_A(x), \mu_A(y)\}$ for all $x, y \in \mathbb{R}$ and all $\lambda \in [0, 1]$.
- A_α is a closed interval of real numbers for every $\alpha \in (0,1]$,
- The support of A which is $supp(A) = \overline{\{x \in \mathbb{R}: \mu_A(x) > 0\}}$ is bounded where $\overline{K}$ denotes the closure of a subset $K \subseteq \mathbb{R}$ in the usual topology of $\mathbb{R}$.

Definition 3 (Fuzzy logic system) [63]: A fuzzy logic system (FLS) includes three parts: fuzzification, fuzzy inference engine and defuzzification. In the fuzzification process, the fuzzy sets are formed for all input variables. The fuzzy inference engine takes into account the input variables and the logical relations between them and uses fuzzy logic operations to generate the output. In the defuzzification process, the output fuzzy set is converted into a crisp value.

When the service provider receives a request from an existing consumer and QoS parameters are different from the previous SLA, then the service provider needs to rely on some parameters for decision-making. These parameters are proposed as follows:

- T_{trend} which is the transaction trend of a consumer with available QoS parameters as defined previously in Eq. 1.
- N which is the similarity of the SLO parameter between a new request and an existing record.
- T_{knn} which is the weighted average of T_{trend} of top-K nearest neighbours requesting consumer with the same SLO parameters as requested by the consumer. T_{knn} has a significant impact on decision-making [18, 21, 43]. The T_{knn} is calculated by considering the enhanced top-K nearest neighbours of a requesting consumer, the degree of similarity between a consumer and nearest neighbours and the value of a transaction trend as presented in Equation 3.

$$T_{knn} = \frac{1}{n}\left[\sum_{a=1}^{n} KN_{enh}\,(a)\{PC2(a) * T_{trend}(a)\}\right] \tag{3}$$

where $\boldsymbol{T_{knn}}$ is a transaction trend for all top-K nearest neighbours, (a) are the nearest neighbours that start at 1 and move to n, which is the total number of top-K nearest neighbours, KN_{enh} (a) is the a^{th} enhanced top-K nearest neighbour, *PC2(a)* is the Pearson correlation coefficient value for a^{th} nearest neighbours, and T_{trend} (a) is the transaction trend for a^{th} nearest neighbours.

The parameters A, B and C are taken as inputs by the CMRADS model to generate an output (i.e. decision variable (*D*)) that solves the decision-making problem which is to accept or reject the marginal resource request.

3.2.1. Fuzzy Membership Functions

This paper assumes that parametric triangular and trapezoidal membership function because these are good enough to capture the vagueness of parameters. The membership functions of the linguistic terms for the input and output variables are based on a combination of triangular and trapezoidal fuzzy numbers to increase the sensitivity in some bounds. The universe discourse of T_{trend} is considered as $[0, 100]$ and partitioned by three linguistic variables: below (B), same (S), and above (A). The membership functions of T_{trend} are defined as follows and are also presented in Figure 3 and Equation 4, 5 and 6.

$$\mu_{T_{trend}(B)}(x) = \begin{cases} 1 & x \le 30 \\ (40-x)/10 & 30 < x \le 40 \end{cases} \tag{4}$$

$$\mu_{T_{trend}(S)}(x) = \begin{cases} (x-30)/10 & 30 \le x < 40 \\ 1 & 40 \le x < 60 \\ (70-x)/10 & 60 \le x < 70 \end{cases} \tag{5}$$

$$\mu_{T_{trend}(A)}(x) = \begin{cases} (x-60)/10 & 60 \le x < 70 \\ 1 & x \ge 70 \end{cases} \tag{6}$$

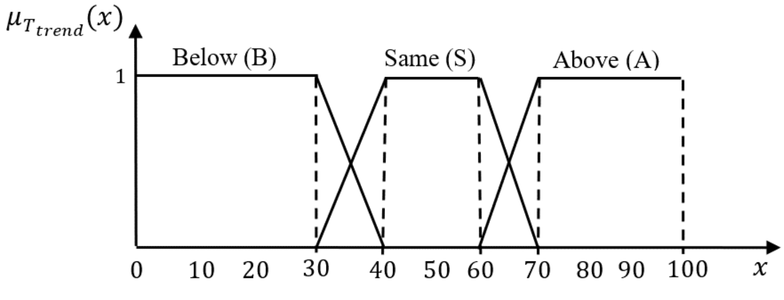


Figure 3: Membership function of T_{trend}.

The second input of the model is N which shows the degree of similar SLO parameters for a requesting consumer between its current request and an existing record. For the sake of simplicity, the arithmetic average is adopted here, as presented in Equation 7:

$$N = \frac{Number\ of\ repeated\ SLOs}{Total\ number\ of\ requested\ SLOs} \tag{7}$$

The universe of discourse of *N* is defined as $[0,1]$, and the membership functions for this input are as shown in Figure 4, Equation 8, 9, 10 and 11 and are also described as follows by four linguistic variables, namely none (N), minimal (Mi), partial (P), and maximal (Ma):

$$\mu_{N(N)}(x) = \begin{cases} 1 & x = 0 \\ (0.01 - x)/0.01 & 0 < x \leq 0.01 \end{cases} \tag{8}$$

$$\mu_{N(Mi)}(x) = \begin{cases} x/0.01 & 0 \leq x \leq 0.01 \\ (0.5 - x)/0.49 & 0.01 < x \leq 0.5 \end{cases} \tag{9}$$

$$\mu_{N(P)}(x) = \begin{cases} (x - 0.01)/0.49 & 0.01 \leq x \leq 0.5 \\ (1 - x)/0.5 & 0.5 < x \leq 1 \end{cases} \tag{10}$$

$$\mu_{N(Ma)}(x) = \begin{cases} (x - 0.5)/0.5 & 0.5 \leq x < 1 \\ 1 & x = 1 \end{cases} \tag{11}$$

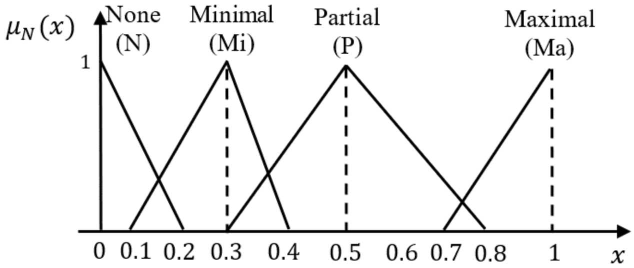


Figure 4: Membership function of *N*.

The last input is ET_{knn}which is the enhanced top-K nearest neighbours of a requesting consumer that has previously used the same SLOs as requested by a consumer in a new request [59]. To select the nearest neighbour (NN) of a consumer, we use a user-based collaborative filtering method (UCF) [64, 65] to select those consumers from a database that have similar profile pattern, which indicates that they have used similar SLOs as those requested by a consumer as presented in Equation 12 [66]:

$$NN = \frac{\sum_i^n [sim\ (p,q_i) * rs_i \{n_i \in N | rs \in R\}]}{n} \tag{12}$$

where *p* is requesting consumer, *q* is the selected nearest neighbours of p, N is a set of all neighbours, *rs* is set of services with the same QoS parameters as those requested by a consumer for a new service, *n* is the total number of services/resources and $sim\ (p, q_i)$ is the similarity between a consumer and all existing similar consumers based on the Pearson correlation coefficient (PCC), as presented in Equation 13:

$$Sim(p,q) = \frac{n(\sum pq) - (\sum p)(\sum q)}{\sqrt{[n \sum p^2 - (\sum p)^2][n \sum q^2 - (\sum q)^2]}} \tag{13}$$

In the literature [28, 67-69], we observed that the PCC provides good prediction accuracy in different regression and recommender systems. The PCC method takes the numeric value ranges from -1 to 1, where -1 and 1 denote a negative and positive linear relationship, respectively and 0 denotes no linear relationship. The traditional top-KNN selection method includes all neighbours that have the PCC value ranges from -1 to +1, however, we have seen [59] that the prediction accuracy significantly decreases by considering those neighbours that have a negative PCC value, because those neighbours have very limited similarities. To overcome this issue, we propose an enhanced top-KNN selection method that only selects those neighbours that have a positive linear relationship with a requesting consumer, as presented in Equation 14.

$$ET_{knn}(p) = \{p_s | p_s \in T_k(p), Sim(p_s, p) > 0, p_s \neq p\} \tag{14}$$

We categorize the nearest neighbours into three classes representing by three linguistic variables, namely below (B), same (S), and above (A) in a universe of discourse of $[0,100]$ and use the same membership function defined for T_{trend}.

The output variable D corresponds to the decision result in which a consumer request is approved. The universe of discourse of D is defined as $[0,1]$ as presented in Figure 5, and Equation 15 and 16. The membership function is as follows:

$$\mu_{D(R)}(x) = \begin{cases} 1 & 0 \le x < 0.2 \\ (0.8 - x)/0.6 & 0.2 \le x \le 0.8 \end{cases} \tag{15}$$

$$\mu_{D(A)}(x) = \begin{cases} (x - 0.2)/0.6 & 0.2 \le x < 0.8 \\ 1 & 0.8 \le x \le 1 \end{cases} \tag{16}$$

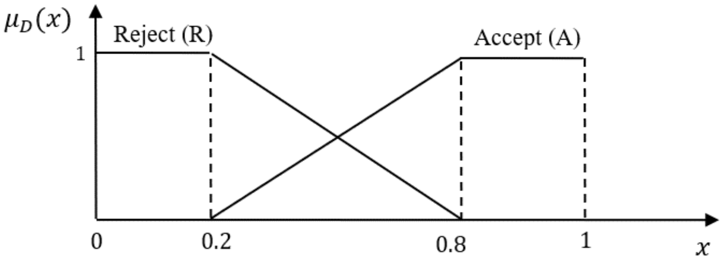


Figure 5: Membership function of *D*.

3.2.2. Fuzzy Rules

The logical relations between input variables (i.e. T_{trend}, N, and ET_{knn}) and the output variable (i.e. D) are demonstrated by 36 fuzzy rules, because conditions 3 *4*3 can cover all scenarios, as shown in Table 2. To define these rules, the authors' knowledge of and experience in working with cloud data centers and their consultation with a cloud service provider have been relied upon.

Table 2: Fuzzy rules.

Rule		T_{trend}		N		ET_{knn}		D
1.	If	A	and	N	and	A	then	A
2.	If	A	and	N	and	S	then	A
3.	If	A	and	N	and	B	then	R
4.	If	A	and	Mi	and	A	then	A
5.	If	A	and	Mi	and	S	then	A
6.	If	A	and	Mi	and	B	then	R
7.	If	A	and	P	and	A	then	A
8.	If	A	and	P	and	S	then	A
9.	If	A	and	P	and	B	then	A
10.	If	A	and	Ma	and	A	then	A
11.	If	A	and	Ma	and	S	then	A
12.	If	A	and	Ma	and	B	then	A
13.	If	S	and	N	and	A	then	A
14.	If	S	and	N	and	S	then	A
15.	If	S	and	N	and	B	then	R
16.	If	S	and	Mi	and	A	then	A
17.	If	S	and	Mi	and	S	then	A

18.	If	S	and	Mi	and	B	then	R
19.	If	S	and	P	and	A	then	A
20.	If	S	and	P	and	S	then	A
21.	If	S	and	P	and	B	then	R
22.	If	S	and	Ma	and	A	then	A
23.	If	S	and	Ma	and	S	then	A
24.	If	S	and	Ma	and	B	then	A
25.	If	B	and	N	and	A	then	A
26.	If	B	and	N	and	S	then	A
27.	If	B	and	N	and	B	then	R
28.	If	B	and	Mi	and	A	then	A
29.	If	B	and	Mi	and	S	then	A
30.	If	B	and	Mi	and	B	then	R
31.	If	B	and	P	and	A	then	A
32.	If	B	and	P	and	S	then	R
33.	If	B	and	P	and	B	then	R
34.	If	B	and	Ma	and	A	then	R
35.	If	B	and	Ma	and	S	then	R
36.	If	B	and	Ma	and	B	then	R

3.2.3. Fuzzy Inference Engine

There are several inference methods; however, the most commonly used methods in the fuzzy community are Mamdani [70] and Takagi and Sugeno [71]. This paper uses Mamdani's method as it is widely accepted for capturing expert knowledge and it allows expertise to be described in a more intuitive, human-like manner. Furthermore, Sugeno's method uses weighted average to compute the crisp output, whereas Mamdani's method has an expressive power and interpretability output [72]. Table 3 lists the characteristics of Mamdani's model that are used to implicate every single rule and aggregate the outcomes from all rules into a single output fuzzy set. In the defuzzification process, the output fuzzy set of decision variables (i.e. *D*) is converted into a crisp value, which is used for decision-making.

Table 3: Mamdani's Fuzzy Model [62].

Operation	Operator	Formula
Union (OR)	MAX	$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x)$
Intersection (AND)	MIN	$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x)$
Implication	MIN	$min(\mu_A(x), \mu_B(x))$
Aggregation	MAX	$max(min(\mu_A(x), \mu_B(x)))$
Defuzzification	CENTROID	$COA = Z^* = \frac{\int z\, \mu_C(z)\, dz}{\int \mu_C(z)\, dz}$

$\mu_C(x)$ = value of the resultant membership function

$\mu_A(x)$ = value of the membership function where the input belongs to the fuzzy set A

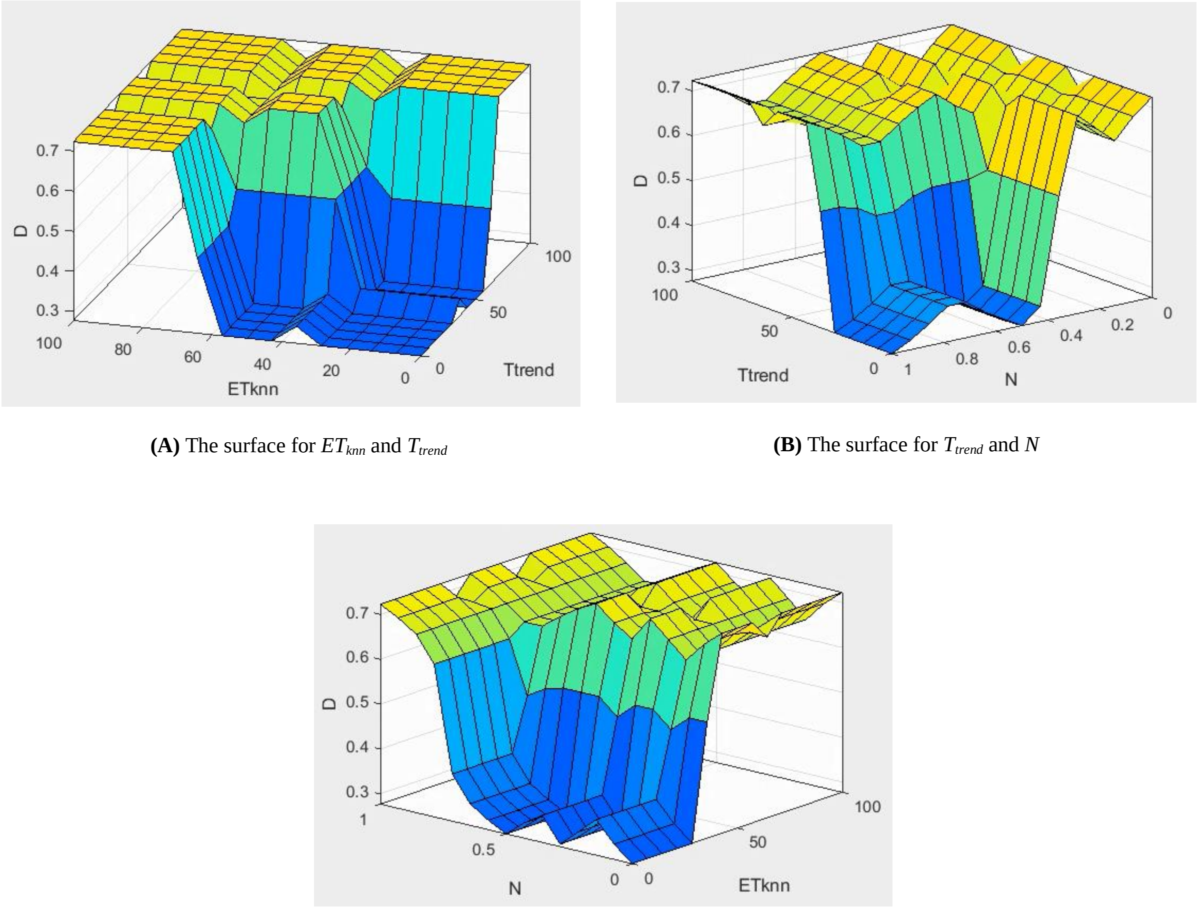
z = abscissa value, $\mu_C(z)$ is the ordinate

4. Implementation and Evaluation

In this section, we present the performance and evaluation of our proposed model for deciding on marginal resource allocation through a semi-real case study.

4.1. Experiment Setup

The model is implemented in MATLAB R2018a on a VMware Horizon Client. Figure 6 presents the control surface of the output variable for every two input variables. As can be seen from Figure 6.A, when the values of the ET_{knn} and T_{trend} are less than 50%, the D output is close to zero representing a reject decision. As can be seen from Figure 6.C, it is evident that when the similarity of SLA is lower than 0.6 and ET_{knn} is less than 45%, the decision variable is almost a reject decision. However, when the similarity of SLA is bigger than 0.6 and T_{trend} is less than 40%, the output variable is a reject decision as inferred from Figure 6.B.



(A) The surface for ET_{knn} and T_{trend}

(B) The surface for T_{trend} and N

(C) The surface for N and ET_{knn}

Figure 6: The proposed model surfaces.

In addition, the model incorporates a time series dataset [73] that comprises 60 time intervals and considers the QoS parameters of throughput, response time and availability. The dataset consists of 142 consumers using 4532 web services; however, for the sake of simplicity, data on only one single web service are fed into the model.

4.2. Case Study

The company opened its first store, a Japanese home center, in 1969, and now operates a total of 144 stores these being primarily home centers, as well as drugstores, book stores, and other retail locations in the Kinki and Chugoku regions of Japan. Although its data-center based store system servers and the PCs within the stores are managed together online, the company began transitioning some of their systems to the cloud three years ago. The company business systems, which had been split into human resources, documents, and client management, are now managed and distributed to two cloud environments using the VMware-based cloud called ASPIRE to create a system with high availability that provides work applications to each location. The company delivers applications to employee’s PCs using client operations management software, and although it is able to manage security and logs, with roughly 600 computers company-wide, the operational costs have been increasing recently.

The nodes and cloud configuration are presented in Figure 7. The business systems are in a cloud environment with the SmartVPN connected to two ASPIRE environments along with connections to the Internet. This allows the other systems to be normally connected to even if one of them is experiencing a heavy load. The Internet can also be connected to any of the systems without any effect on external incoming and outgoing e-mail, and the environment has no effect on practical operations.

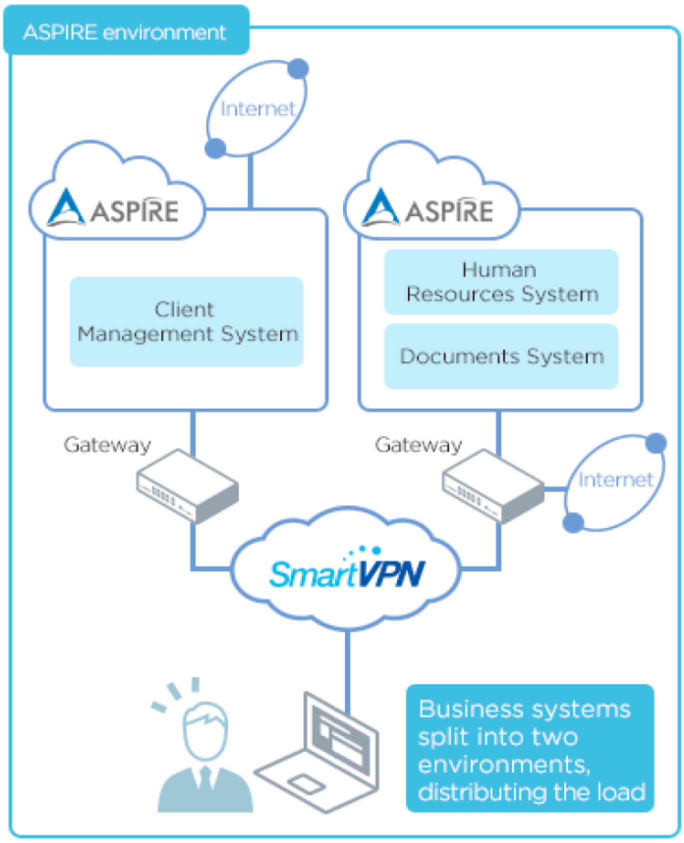


Figure 7: The case study cloud environment.

4.3. Scenario

The company is concerned about the possibility of insufficient bandwidth in the near future. There are currently two issues: 1) server resources and access line bandwidth are stretched thin on the existing cloud system, 2) the client wants an infrastructure that will withstand severe load increases caused by Windows updates, etc. Therefore, the company is reviewing the server configuration and network bandwidth and would like to request a 1Gbps connection to improve the network bandwidth.

The company currently uses availability, durability and latency for both human resource and document systems. In addition, availability, durability, and throughput are used for the client management system. To deal with the bandwidth concern, the company requests a 1Gbps bandwidth for both clouds. Therefore, the scenario represents the decision-making situation in which the company requests a different marginal resource to be added to its set of current marginal resources.

4.4. Results

Table 4 summarizes the company transactions per month for each marginal resource. Using Equation 3, T_{trend} is calculated. As can be seen, the total number of transactions i.e. T_{total} is 439 and the number of transactions involving marginal resources, i.e. T_{succ}, is 337. Therefore, T_{trend} is 76%.

Table 4: The usage of marginal resources.

Resource	No. of transactions without using marginal resources	No. of transactions while using marginal resources	Total number of transactions	T_{trend}= (T_{Succ}/T_{Total})*100
Availability	31	89	120	(337/439)*100=76%
Durability	19	66	85	
Latency	20	110	130	
Throughput	32	72	104	
		T_{succ}=337	T_{total}=439	

Following the model sequence, the variable N is calculated using Equation 5. In the new situation, the company requests five SLOs while four SLOs are repeated. Therefore, N is 0.8. Then, ET_{knn} is calculated, which is the enhanced top-K nearest neighbours of the company that have previously used the same SLOs i.e. availability, durability, latency, throughput and bandwidth. All the nearest neighbors which have previously used bandwidth and have the maximum similarity to the case study are selected. To achieve this, the user-based collaborative (UCF) filtering algorithm which is a non-parametric method and is widely used for classification and regression problems is executed. The UCF algorithm is trained on a large dataset and filters records which involve collaboration with other consumers that have similar characteristics and have requested the same SLO in previous SLAs. To increase the prediction accuracy, all those neighbors that have a positive PCC value are selected, resulting in 20 neighbors. The prediction accuracy of each method is compared using RMSE and MAD, as presented in Table 5 and Figures 8 and 9.

Table 5: RMSE and MAD for the top-K nearest neighbours

#	Consumer ID	RMSE	MAD
1	077	0.602856	0.607137
2	224	0.620337	0.635118
3	045	0.793873	0.641546
4	012	0.863031	0.690760
5	088	0.695218	0.579833
6	013	0.735113	0.607698
7	099	0.675138	0.583150
8	066	0.669813	0.550983
9	075	0.667838	0.628847
10	376	0.706508	0.658846
11	022	0.739534	0.681359
12	176	0.766848	0.678578
13	038	0.769589	0.655349
14	185	0.779035	0.688324
15	432	0.789842	0.690885
16	005	0.794246	0.678535
17	254	0.798566	0.676538
18	176	0.803594	0.732052
19	289	0.826001	0.722424
20	049	0.815077	0.741256

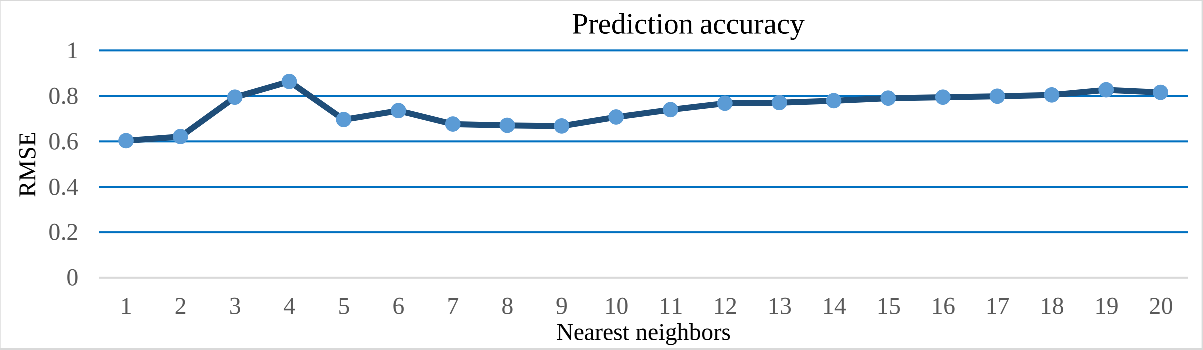


Figure 8: Prediction accuracy of nearest neighbors using RMSE as the benchmark

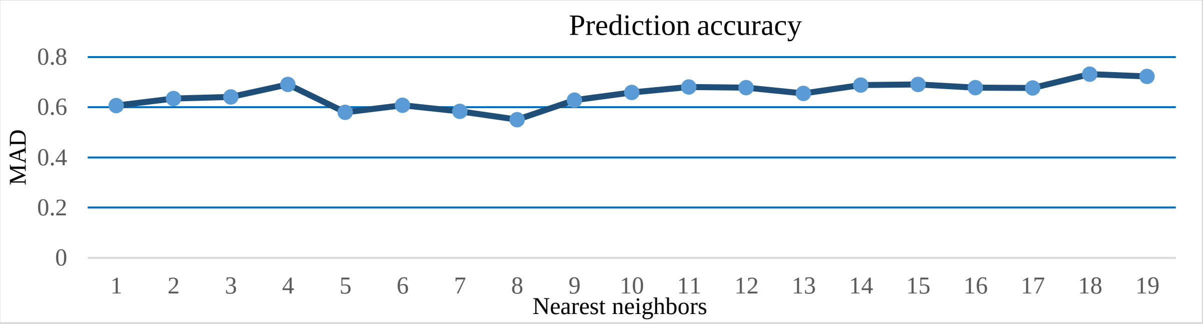


Figure 9: Prediction accuracy of nearest neighbors using MAD as the benchmark

As can be seen from Table 5 and Figures 8 and 9, the prediction accuracy fluctuates up until nine neighbors, but after the ninth neighbor, it decreases continuously until the end of the list. Hence, for this experiment, the top-K is set as 9 and the first nine neighbors are selected. The transaction history for the top-K nearest neighbors and their respective T_{trend} value is presented in Table 6. Therefore, ET_{knn} which is the average of T_{trend} of neighbors is calculated as 53%.

Table 6: The K-nearest neighbours with PCC values and the number of successful and violated transactions.

#	Consumer	PCC	No. of successful transactions	No. of violated transactions	T_{trend} (%)
1	077	0.9999769	28	5	84
2	224	0.9999645	94	48	53
3	045	0.9999413	59	12	83
4	012	0.9999371	16	25	39
5	088	0.9997480	1	3	25
6	013	0.9996665	23	8	74
7	099	0.9993055	10	5	66
8	066	0.9992993	8	6	57
9	075	0.9992344	0	1	0
					ET_{knn} =481/9=53

The model takes into account the variables and using the fuzzy rules presented in Table 2 and Mamdani's fuzzy operations, the output variable D is calculated as 0.87 which linguistically represents the "accept" decision. A sensitivity analysis is also done on the uncertain variable of ET_{knn} with 10% increase and decrease in the value, the decision-making variable of D is not affected meaningfully. While the decision-making situation includes three input variables, it can be seen the decision making without the support of the proposed model is not easy. Even for an experienced decision maker, it is hard to recall the knowledge and reason based on these quantitative variables. It is even harder when more qualitative or quantitative variables may be added to the problem in the future. The proposed model also shows its advantage better when the variables are not supporting strongly the acceptance or rejection decision making problem.

4.4. Performance Evaluation and Comparison with MCDM Techniques

The validity of the proposed model is based on its performance and case-by-case investigations. The results were discussed with and assured by the relevant experts. Theoretically, the proposed fuzzy method makes decision making easier by means of linguistic terms and approximate reasoning. It captures the judgments of specialists and stores them in a knowledge base to minimize rough evaluations that lead to suboptimal measurements. It can handle both quantitative data and imprecisely defined qualitative information. Further, the proposed fuzzy method can be extended in practical terms for use with any number of inputs. Moreover, the model provides more informative and reliable analytical results and facilitates decision making in less time.

The cloud marginal resource allocation decision making represents a multi criteria decision making (MCDM) problem in which multiple and conflicting criteria make the decision making complex. Therefore, one may think about using traditional, fuzzy, or hybrid MCDM techniques for making this decision. However, the MCDM techniques have shown some challenges over the years. In comparison with current MCDM-based methods, the proposed model overcomes several challenges. Existing MCDM methods either focus on obtaining the result as a ranking or a utility function to aid decision makers [74]. They typically overlook the relationships among the involved criteria and fail to identify the imprecise reasoning embedded in their criteria with respect to the addressed problem. These MCDM methods assume that the criteria are independent and hierarchical in structure. However, the relationships among criteria are usually interdependent with certain feedback effects. To identify the interrelated relationships among the variables, the decision making trial and evaluation laboratory (DEMATEL) technique can build an influential network relations matrix to find the influential weights of DEMATEL-based analytic network process (DANP). This technique can model some, but not all, of the interdependent and feedback relationships among the criteria. In addition, in MCDM methods like fuzzy ANP, the interdependence among the factors must be analyzed first to reduce the number of pairwise comparisons, which is one of its most often-mentioned disadvantages [75].

Despite its benefits, the proposed method has some limitations. It relies solely on fuzzy rules elicited from experts. Further, tuning the fuzzy rules through machine learning could improve the performance of this method with historical data, which may be addressed in future work.

5. Conclusion and Future Work

In the cloud environment, SLAs have become the main criterion for service selection. The elastic characteristic of cloud computing enables the cloud consumer to request resources and services without worrying about its scalability. Consumers must enter into an SLA contract to ensure the provision of the requested resources and the on-time delivery of services within the agreed QoS parameters. Some consumers need additional resources on top of those already requested, with the same QoS parameters. For small and medium-sized cloud service providers, it is vital that they manage their resources wisely to avoid any violation and penalties. This paper has presented the CMRADS model to assist cloud providers to manage their resources optimally and to avoid violations and penalties due to a lack of resources. The model covers all possible scenarios, such as whether a particular consumer is new or not, and if the requests for marginal resources are for the same set of SLOs with the same QoS parameters, throughput, response time and availability and with different QoS parameters, durability and latency. The model depends on the capabilities of the user-based collaborative filtering method with the top-k nearest neighbour algorithm and a fuzzy logic system to make a decision. The enhanced top-k nearest neighbour algorithm is used for clustering the consumers and generating a reputation value in comparison with a threshold value that determines the decision. The fuzzy logic system includes several fuzzy variables and rules and the Mamdani inference engine for decision-making.

In our future work, we will find and analyse the hidden arrays between different SLOs and different low-level metrics and how these parameters assist the cloud provider in predicting and managing resources. In addition, we will apply the approach for large-scale cloud providers using a real cloud dataset.

References

1. Gartner, *Gartner Forecasts Worldwide Public Cloud Revenue to Grow 17.5 Percent in 2019*. 2019, Gartner: Stratford, Connecticut, USA.
2. Statista, *Spending on public cloud IT services (SaaS/PaaS) worldwide 2015-2020*. 2019: Hamburg, Germany.
3. Hussain, W., F.K. Hussain, and O.K. Hussain. *Maintaining Trust in Cloud Computing through SLA Monitoring*. in *Neural Information Processing*. 2014. Springer.
4. Son, S., et al., *Adaptive trade-off strategy for bargaining-based multi-objective SLA establishment under varying cloud workload.* The Journal of Supercomputing, 2016. **72**(4): p. 1597-1622.
5. Silaghi, G.C., L.D. ŞErban, and C.M. Litan, *A time-constrained SLA negotiation strategy in competitive computational grids.* Future Generation Computer Systems, 2012. **28**(8): p. 1303-1315.
6. Gwak, J. and K.M. Sim, *A novel method for coevolving PS-optimizing negotiation strategies using improved diversity controlling EDAs.* Applied intelligence, 2013. **38**(3): p. 384-417.
7. Sim, K.M., *Grid resource negotiation: survey and new directions.* IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2010. **40**(3): p. 245-257.
8. Gao, H., et al., *Toward service selection for workflow reconfiguration: An interface-based computing solution.* 2018. **87**: p. 298-311.
9. Abts, D. and B.J.Q. Felderman, *A guided tour through data-center networking.* 2012. **10**(5): p. 10.
10. Luong, N.C., et al., *Resource management in cloud networking using economic analysis and pricing models: A survey.* 2017. **19**(2): p. 954-1001.
11. Aazam, M. and E.-N. Huh. *Advance resource reservation and QoS based refunding in cloud federation*. in *2014 IEEE Globecom Workshops (GC Wkshps)*. 2014. IEEE.
12. Shen, H., Z.J.I.T.o.P. Li, and D. Systems, *New bandwidth sharing and pricing policies to achieve a win-win situation for cloud provider and tenants.* 2016. **27**(9): p. 2682-2697.

13. Prasad, K.H., et al. *Resource allocation and SLA determination for large data processing services over cloud*. in *2010 IEEE International Conference on Services Computing*. 2010. IEEE.

14. Altmann, J. and M.M.J.F.G.C.S. Kashef, *Cost model based service placement in federated hybrid clouds.* 2014. **41**: p. 79-90.

15. Shepherd, W.G., *Ramsey pricing: Its uses and limits.* Utilities Policy, 1992. **2**(4): p. 296-298.

16. Hadji, M. and D.J.I.t.o.c.c. Zeghlache, *Mathematical programming approach for revenue maximization in cloud federations.* 2017. **5**(1): p. 99-111.

17. SM, D.K., N. Sadashiv, and R. Goudar. *Priority based resource allocation and demand based pricing model in peer-to-peer clouds*. in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. 2014. IEEE.

18. Hussain, W., et al., *Provider-based optimized personalized viable SLA (OPV-SLA) framework to prevent SLA violation*. 2016, British Computer Society.

19. Hussain, W., F.K. Hussain, and O.K. Hussain. *SLA Management Framework to Avoid Violation in Cloud*. in *International Conference on Neural Information Processing*. 2016. Springer.

20. Hussain, W., et al., *Provider-based optimized personalized viable SLA (OPV-SLA) framework to prevent SLA violation.* The Computer Journal, 2016.

21. Hussain, W., F.K. Hussain, and O. Hussain, *Comparative analysis of consumer profile-based methods to predict SLA violation*, in *FUZZ-IEEE*, IEEE, Editor. 2015, IEEE: Istanbul Turkey.

22. Emeakaroha, V.C., et al. *Low level metrics to high level SLAs-LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments*. in *High Performance Computing and Simulation (HPCS), 2010 International Conference on*. 2010. IEEE.

23. Emeakaroha, V.C., et al., *Towards autonomic detection of SLA violations in Cloud infrastructures.* Future Generation Computer Systems, 2012. **28**(7): p. 1017-1029.

24. Zhang, Y., Z. Zheng, and M.R. Lyu. *Exploring latent features for memory-based QoS prediction in cloud computing*. in *Reliable Distributed Systems (SRDS), 2011 30th IEEE Symposium on*. 2011. IEEE.

25. Kamel, A., A. Al-Fuqaha, and M. Guizani, *Exploiting client-side collected measurements to perform QoS assessment of IaaS.* IEEE Transactions on Mobile Computing, 2015. **14**(9): p. 1876-1887.

26. Redl, C., et al. *Automatic SLA matching and provider selection in grid and cloud computing markets*. in *Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing*. 2012. IEEE Computer Society.

27. Joshi, K.P. and C. Pearce. *Automating cloud service level agreements using semantic technologies*. in *Cloud Engineering (IC2E), 2015 IEEE International Conference on*. 2015. IEEE.

28. Hussain, W., et al., *Comparing time series with machine learning-based prediction approaches for violation management in cloud SLAs.* 2018. **89**: p. 464-477.

29. Dastjerdi, A.V., et al., *CloudPick: a framework for QoS-aware and ontology-based service deployment across clouds.* Software: Practice and Experience, 2015. **45**(2): p. 197-231.

30. Gao, H., et al., *Applying improved particle swarm optimization for dynamic service composition focusing on quality of service evaluations under hybrid networks.* 2018. **14**(2): p. 1550147718761583.

31. Haq, I.U., I. Brandic, and E. Schikuta, *Sla validation in layered cloud infrastructures*, in *Economics of Grids, Clouds, Systems, and Services*. 2010, Springer. p. 153-164.

32. Yin, Y., L. Chen, and J.J.I.A. Wan, *Location-Aware Service Recommendation With Enhanced Probabilistic Matrix Factorization.* 2018. **6**: p. 62815-62825.

33. Yin, Y., et al., *Network location-aware service recommendation with random walk in cyber-physical systems.* 2017. **17**(9): p. 2059.

34. Yin, Y., et al., *QoS Prediction for Web Service Recommendation with Network Location-Aware Neighbor Selection.* International Journal of Software Engineering and Knowledge Engineering, 2016. **26**(04): p. 611-632.

35. Yin, Y., et al., *QoS Prediction for Service Recommendation with Deep Feature Learning in Edge Computing Environment.* Mobile Networks and Applications, 2019.

36. Romano, L., et al. *A novel approach to QoS monitoring in the cloud*. in *Data Compression, Communications and Processing (CCP), 2011 First International Conference on*. 2011. IEEE.

37. Cicotti, G., et al., *How to monitor QoS in cloud infrastructures: the QoSMONaaS approach.* International Journal of Computational Science and Engineering, 2015. **11**(1): p. 29-45.

38. Cicotti, G., et al. *QoS monitoring in a cloud services environment: the SRT-15 approach*. in *European Conference on Parallel Processing*. 2011. Springer.

39. Leitner, P., et al. *Runtime prediction of service level agreement violations for composite services*. in *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*. 2010. Springer.

40. Ciciani, B., et al. *Automated workload characterization in cloud-based transactional data grids*. in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*. 2012. IEEE.

41. Hussain, W., et al., *Formulating and managing viable SLAs in cloud computing from a small to medium service provider's viewpoint: A state-of-the-art review.* Information Systems, 2017. **71**: p. 240-259.
42. Gao, H., et al., *Applying Probabilistic Model Checking to Financial Production Risk Evaluation and Control: A Case Study of Alibaba's Yu'e Bao.* 2018(99): p. 1-11.
43. Hussain, W., et al., *Profile-based viable Service Level Agreement (SLA) Violation Prediction Model in the Cloud*, in *2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC)*. 2015, IEEE: Krakow, Poland. p. 268-272.
44. Hussain, W., et al., *Provider-Based Optimized Personalized Viable SLA (OPV-SLA) Framework to Prevent SLA Violation.* The Computer Journal, 2016. **59**(12): p. 1760-1783.
45. Mustafa, S., et al., *SLA-Aware Energy Efficient Resource Management for Cloud Environments.* 2018. **6**: p. 15004-15020.
46. Cheetham, W., A. Varma, and K. Goebel. *Case-Based Reasoning at General Electric*. in *FLAIRS Conference*. 2001.
47. Meland, P.H., et al., *Expressing cloud security requirements for slas in deontic contract languages for cloud brokers.* International Journal of Cloud Computing 2, 2014. **3**(1): p. 69-93.
48. Hussain, W., et al., *Comparing time series with machine learning-based prediction approaches for violation management in cloud SLAs.* Future Generation Computer Systems, 2018. **89**: p. 464-477.
49. Hussain, W., et al., *Risk-based framework for SLA violation abatement from the cloud service provider's perspective.* The Computer Journal, 2018.
50. Brandic, I., et al. *Laysi: A layered approach for sla-violation propagation in self-manageable cloud infrastructures*. in *Computer Software and Applications Conference Workshops (COMPSACW), 2010 IEEE 34th Annual*. 2010. IEEE.
51. Emeakaroha, V.C., et al. *Casvid: Application level monitoring for sla violation detection in clouds*. in *Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual*. 2012. IEEE.
52. Mosallanejad, A. and R. Atan, *HA-SLA: A Hierarchical Autonomic SLA Model for SLA Monitoring in Cloud Computing.* Journal of Software Engineering and Applications, 2013. **6**(03): p. 114.
53. Katsaros, G., et al., *A Self-adaptive hierarchical monitoring mechanism for Clouds.* Journal of Systems and Software, 2012. **85**(5): p. 1029-1041.
54. Sun, Y., et al. *SLA detective control model for workflow composition of cloud services*. in *Computer Supported Cooperative Work in Design (CSCWD), 2013 IEEE 17th International Conference on*. 2013. IEEE.
55. Cardellini, V., et al. *Sla-aware resource management for application service providers in the cloud*. in *Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on*. 2011. IEEE.
56. Schmieders, E., et al., *Combining SLA prediction and cross layer adaptation for preventing SLA violations.* 2011.
57. Noor, T.H. and Q.Z. Sheng, *Trust as a service: a framework for trust management in cloud environments*, in *Web Information System Engineering–WISE 2011*. 2011, Springer. p. 314-321.
58. Fan, W. and H. Perros. *A reliability-based trust management mechanism for cloud services*. in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*. 2013. IEEE.
59. Hussain, W., F.K. Hussain, and O.K. Hussain. *Comparative analysis of consumer profile-based methods to predict SLA violation*. in *Fuzzy Systems (FUZZ-IEEE), 2015 IEEE International Conference on*. 2015. IEEE.
60. Hussain, W., F. Hussain, and O. Hussain. *Allocating optimized resources in the cloud by a viable SLA model*. in *Fuzzy Systems (FUZZ-IEEE), 2016 IEEE International Conference on*. 2016. IEEE.
61. Hussain, W., F.K. Hussain, and O.K. Hussain. *Risk Management Framework to Avoid SLA Violation in Cloud from a Provider's Perspective*. in *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*. 2016. Springer.
62. Naderpour, M., J. Lu, and G. Zhang, *An intelligent situation awareness support system for safety-critical environments.* Decision Support Systems, 2014. **59**: p. 325-340.
63. Markowski, A.S., et al., *Application of fuzzy logic to explosion risk assessment.* Journal of Loss Prevention in the Process Industries, 2011. **24**(6): p. 780-790.
64. Breese, J.S., D. Heckerman, and C. Kadie. *Empirical analysis of predictive algorithms for collaborative filtering*. in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. 1998. Morgan Kaufmann Publishers Inc.
65. Herlocker, J.L., et al. *An algorithmic framework for performing collaborative filtering*. in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. 1999. ACM.
66. Wang, J., A.P. De Vries, and M.J. Reinders. *Unifying user-based and item-based collaborative filtering approaches by similarity fusion*. in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 2006. ACM.
67. Zhou, H., et al., *A new sampling method in particle filter based on Pearson correlation coefficient.* 2016. **216**: p. 208-215.
68. Lin, L.I.-K., *A concordance correlation coefficient to evaluate reproducibility.* 1989: p. 255-268.

69. Adler, J. and I.J.C.P.A. Parmryd, *Quantifying colocalization by correlation: the Pearson correlation coefficient is superior to the Mander's overlap coefficient.* 2010. **77**(8): p. 733-742.

70. Mamdani, E.H., *Application of fuzzy logic to approximate reasoning using linguistic synthesis.* IEEE Transactions on Computers, 1977. **C-26**(12): p. 1182-1191.

71. Nazir, S., S. Colombo, and D. Manca, *Testing and analyzing different training methods for industrial operators: an experimental approach*, in *Computer Aided Chemical Engineering*, K. Andrzej and T. Ilkka, Editors. 2013, Elsevier. p. 667-672.

72. Kaur, A., A.J.I.j.o.s.c. Kaur, and engineering, *Comparison of mamdani-type and sugeno-type fuzzy inference systems for air conditioning system.* 2012. **2**(2): p. 323-325.

73. Zhang, Y., Z. Zheng, and M.R. Lyu. *WSPred: A time-aware personalized QoS prediction framework for Web services*. in *Software Reliability Engineering (ISSRE), 2011 IEEE 22nd International Symposium on*. 2011. IEEE.

74. Sohaib, O., et al., *Cloud Computing Model Selection for E-commerce Enterprises Using a New 2-tuple Fuzzy Linguistic Decision-Making Method.* 2019.

75. Tzeng, G.-H. and K.-Y. Shen, *New concepts and trends of hybrid multiple criteria decision making*. 2017: CRC Press.